Sequential Monte Carlo Filtering with Gaussian Mixture Sampling

Sehyun Yun[1] and Renato Zanetti[2]
The University of Texas at Austin, Austin, Texas 78712

I. Introduction

Bayesian stochastic estimation of nonlinear and non-Gaussian dynamical systems using sequential Monte Carlo methods continues to receive considerable attention in the literature [1–3]. The Kalman filter provides an exact solution to the minimum mean-square error estimation problem for linear systems corrupted by additive Gaussian noise [4]. However, in practice, the conditions for optimality of the Kalman filter are easily and often violated. The extended Kalman filter (EKF) is a nonlinear (non-optimal) approximation of the optimal Kalman filter that can be applied to nonlinear systems using the same Kalman filtering framework [5]. The possible divergence of the EKF estimates due to severe nonlinearities is a drawback of this procedure. Other linear estimators of nonlinear systems include algorithms that rely on a set of deterministic regression points [6], such as the quadrature Kalman filter [7], the unscented Kalman filter (UKF) [8], and the cubature Kalman filter [9]. These algorithms employ the Gaussian approximation and statistical linearization of the nonlinear functions through a set of regression points. However, these methodologies are not always feasible for very high nonlinearities when the state's probability density function (PDF) is multimodal or very non-Gaussian.

The Gaussian sum filter (GSF) is a nonlinear estimator for nonlinear systems [10, 11]. It is able to account for large deviations from Gaussianity and accommodate multi-modal distributions by approximating the non-Gaussian PDF as a Gaussian Mixture Model (GMM). The GSF includes one linear estimator, such as EKF or UKF, for each of the GMM components. The GSF works best when enough components are taken,

[1] Ph.D. student, Department of Aerospace Engineering and Engineering Mechanics
[2] Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics

each of which with a small enough covariance matrix such that the nonlinear functions can accurately be linearized in the support of each of the components. Much recent research exists in improving the original GSF algorithm to better adapt to, and account for, nonlinearities [12–18].

While GSFs approximate the PDFs as a sum of Gaussians, sequential Monte Carlo methods approximate them by discretization using a finite number of random samples. Monte Carlo methods need to draw from the actual distributions, which are often arduous to obtain; Sequential Importance Sampling (SIS) algorithms, on the other hand, sample from an importance sampling distribution and adjust the weights of each sample accordingly. Particle Filters are a family of SIS algorithms that include a resampling step to mitigate particle (i.e. sample) degeneracy [19]. One of the most popular algorithms chooses the importance distribution as the transition distribution, the so-called Bootstrap Particle filter (BPF) [20]. One possible drawback of the BPF is that it does not directly account for the value of the measurement in the sampling distribution. The Auxiliary Particle Filter mitigates this issue by using an auxiliary variable to account for the value of the measurement in the importance distribution [21]. The resampling step is often critical for practical uses of the Particle Filter and is usually done sampling from a discrete distribution. The Regularized Particle Filter draws from a continuous distribution approximation of the PDF [1] by perturbing the particles after resampling to add diversity to the state space. The approach presented in this work contains a new methodology that includes both these improvements: it samples from a continuous distribution that incorporates the contributions of the current measurement.

The particle filter approximates distributions as discrete, i.e. as weighted sum of Dirac deltas. Other approaches to discretize the PDF include deterministic Dirac mixtures with equal weights [22] and to combine particle filters with Gaussian Mixture Models. In Refs. [23] and [24], the authors start from a GMM and at each cycle they resample in a manner similar to a particle filter. Their resampling step is subject to a matrix inequality constraint that insures the covariance of each of the resampled Gaussian components stays below a desired tunable value.

Reference [25] starts from the Gaussian particle filter derived in [26] to build the Gaussian sum particle filter (GSPF). The GSPF is basically a bank of Gaussian particle filters approximating the conditional distributions by weighted Gaussian mixtures. Ref. [27] introduces the Particle Gaussian Mixture Filter (PGMF) and employs an ensemble of randomly sampled states for the propagation of the conditional state probability density. The propagated ensemble is clustered to recover a Gaussian Mixture Model representation of the propagated PDF. Finally, the posterior PDF can be obtained through a GSF update. This approach is somewhat reminiscent of the Regularized Particle Filter, which uses kernel density estimation as the clustering algorithm.

In this paper we propose to always sample from the posterior distribution, never to combine the distribution at the prior time with an importance distribution, as done in SIS. Therefore, the methodology and the algorithms derived in this work are conceptually and practically very different from the GSPF and PGMF. The proposed methodology is also different from the Regularized Particle Filter, since the regularized particle filter employs kernel density estimation on the particles in order to resample from a continuous distribution. In this work we calculate the posterior distribution directly, we do not approximate it via clustering or kernel density estimator starting from samples. We do not utilize sequential importance sampling like in the GSPF. Moreover, no particles propagation and clustering occurs like in the PGMF, rather an initial GMM is generated at each cycle and from it a posterior PDF is obtained. In addition to the main result, two modifications of the baseline algorithm are proposed to further improve its accuracy. First, an importance sampling version of the algorithm is developed. Then, in the second modification, the initial covariance of the GMM components is not set to zero, but to a small value that removes the bias in the sample covariance.

The remainder of the paper is organized as follows. First the GSF and particle filter algorithms are described. Then, the new algorithms are introduced in section III. In section IV, simulation results using the proposed algorithm are presented followed by some concluding remarks on the methodology and results.

II. Preliminary Notions

The Gaussian Sum Filter (GSF) and Particle Filter (PF) are two common solutions to the nonlinear Bayesian estimation problem and they are briefly reviewed in this section.

A. Gaussian Mixture Models and the Gaussian Sum Filter

Throughout this paper we consider general discrete-time nonlinear dynamics and measurements. The dynamics is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{\nu}_k) \tag{1}$$

where $\boldsymbol{f}_k$ is some non-linear function and the process noise $\boldsymbol{\nu}_k$ is a zero-mean, white sequence, independent from the initial distribution of $\boldsymbol{x}_0$ and possessing covariance matrix $\boldsymbol{Q}_k$. The measurement is

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{x}_k) + \boldsymbol{\eta}_k \tag{2}$$

the measurement noise $\boldsymbol{\eta}_k$ is a zero-mean, white sequence with covariance matrix $\boldsymbol{R}_k$, independent from all other random quantities.

The GSF approximates the conditional PDF by combining several Gaussian components having different means and covariance matrices, and this approximation of the probability distribution is called a Gaussian Mixture Model (GMM). The conditional PDF of $\boldsymbol{x}_k|\boldsymbol{y}_1...\boldsymbol{y}_k$ is expressed as follows:

$$p_{\boldsymbol{x}_k}(\boldsymbol{x}_k) = \sum_{i=1}^{N} \omega_{k|k}^{(i)}\, n(\boldsymbol{x}_k;\, \boldsymbol{\mu}_{k|k}^{(i)}, \boldsymbol{P}_{k|k}^{(i)}) \tag{3}$$

where $n(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{P})$ represents the Gaussian pdf with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{P}$, and $\omega_{k|k}^{(i)}$, $\boldsymbol{\mu}_{k|k}^{(i)}$ and $\boldsymbol{P}_{k|k}^{(i)}$ are the weights, means, and covariance matrices of the i-th Gaussian component. The PDF's normalization and positivity properties lead to the following constraints on the weights

$$\omega_{k|k}^{(i)} \geq 0,\ \forall i \qquad \sum_{i=1}^{N} \omega_{k|k}^{(i)} = 1 \tag{4}$$

(it is actually possible to define some of the weights negative, but that type of GMM approximation is not considered here.) Assuming the covariance matrices are "small" enough (such that linearization of the dynamics and measurements holds in the domain of likely realization of each of the components), then each of the components remains approximately Gaussian at all times and it is propagated and updated using the conventional EKF equations. The time update equations are described as:

$$\boldsymbol{\mu}_{k+1|k}^{(i)} = \boldsymbol{f}_k\big(\boldsymbol{\mu}_{k|k}^{(i)}\big) \tag{5}$$

$$\boldsymbol{P}_{k+1|k}^{(i)} = \boldsymbol{F}_k^{(i)} \boldsymbol{P}_{k|k} \boldsymbol{F}_k^{(i)\mathrm{T}} + \boldsymbol{G}_k^{(i)} \boldsymbol{Q}_k \boldsymbol{G}_k^{(i)\mathrm{T}} \tag{6}$$

$$\omega_{k+1|k}^{(i)} = \omega_{k|k}^{(i)} \tag{7}$$

where $\boldsymbol{F}_k^{(i)}$ and $\boldsymbol{G}_k^{(i)}$ are the Jacobian of the dynamics evaluated at the component's mean $\boldsymbol{x}_x = \boldsymbol{\mu}_{k|k}^{(i)}$ and at $\boldsymbol{\nu}_k = \boldsymbol{0}$, respectively.

The measurement update follows from Bayes's rule and is given by

$$\boldsymbol{\mu}^{(i)}_{k|k} = \boldsymbol{\mu}^{(i)}_{k|k-1} + \boldsymbol{K}^{(i)}_k \Big(\boldsymbol{y}_k - \boldsymbol{h}_k\big(\boldsymbol{\mu}^{(i)}_{k|k-1}\big)\Big) \tag{8}$$

$$\boldsymbol{P}^{(i)}_{k|k} = \boldsymbol{P}^{(i)}_{k|k-1} - \boldsymbol{K}^{(i)}_k \left(\boldsymbol{H}^{(i)}_k \boldsymbol{P}^{(i)}_{k|k-1} \boldsymbol{H}^{(i)\mathrm{T}}_k + \boldsymbol{R}_k\right) \boldsymbol{K}^{(i)\mathrm{T}}_k \tag{9}$$

$$\boldsymbol{K}^{(i)}_k = \boldsymbol{P}^{(i)}_{k|k-1}\, \boldsymbol{H}^{(i)\mathrm{T}}_k \left(\boldsymbol{H}^{(i)}_k \boldsymbol{P}^{(i)}_{k|k-1} \boldsymbol{H}^{(i)\mathrm{T}}_k + \boldsymbol{R}_k\right)^{-1} \tag{10}$$

$$\omega^{(i)}_{k|k} = \frac{\omega^{(i)}_{k|k-1}\, \beta^i_k}{\sum_{i=1}^N \omega^{(i)}_{k|k-1}\, \beta^i_k} \tag{11}$$

where

$$\beta^i_k = n\Big(\boldsymbol{y}_k;\ \boldsymbol{h}_k\big(\boldsymbol{x}^{(i)}_{k|k-1}\big),\ \boldsymbol{H}^{(i)}_k \boldsymbol{P}^{(i)}_{k|k-1} \boldsymbol{H}^{(i)\mathrm{T}}_k + \boldsymbol{R}_k\Big) \tag{12}$$

where $\boldsymbol{H}^{(i)}_k$ is the Jacobian of the measurement evaluated at the prior mean $\boldsymbol{\mu}^{(i)}_{k|k-1}$. The weights are scaled so that they add to one.

Lastly, the total mean $\boldsymbol{\mu}_{k|k}$ and covariance matrix $\boldsymbol{P}_{k|k}$ of the posterior GMM are given by

$$\boldsymbol{\mu}_{k|k} = \sum_{i=1}^N \omega^{(i)}_{k|k}\, \boldsymbol{\mu}^{(i)}_{k|k} \tag{13}$$

$$\boldsymbol{P}_{k|k} = \sum_{i=1}^N \omega^{(i)}_{k|k} \left(\boldsymbol{P}^{(i)}_{k|k} + \boldsymbol{\mu}^{(i)}_{k|k} \boldsymbol{\mu}^{(i)\mathrm{T}}_{k|k} - \boldsymbol{\mu}_{k|k} \boldsymbol{\mu}^{\mathrm{T}}_{k|k}\right) \tag{14}$$

It is noted that the GMM approximation of the conditional PDF approaches to the true PDF under the assumption that there are a sufficient number of Gaussian components and that each of them has covariance matrix small enough such that the linearization of each component around its mean is representative of the nonlinear dynamics and measurements.

B. The Particle Filter

Particle Filters (PF) are a subset of sequential Monte Carlo methods that use Sequential Importance Sampling with Resempling (SISR). The PF approximates the continuous PDF as a discrete probability mass function (PMF), therefore the PDF is composed by a weighted sum of Dirac delta functions [19].

$$p_{\boldsymbol{x}_k}(\boldsymbol{x}_k) \approx \sum_{i=1}^N \omega^{(i)}_{k|k}\, \delta(\boldsymbol{x}_k - \boldsymbol{x}^{(i)}_{k|k}) \tag{15}$$

If it were feasible to compute the actual posterior distribution at the next time step $p_{\boldsymbol{x}_{k+1}}(\boldsymbol{x}_{k+1})$ starting from $p_{\boldsymbol{x}_k}(\boldsymbol{x}_k)$ and to sample from it; then we would use standard Monte Carlo techniques. However, since it is usually unfeasible to sample from the actual posterior distribution, an importance distribution is often used instead. The Bootstrap particle filter uses the transition distribution as the importance distribution. With capital letters we indicate the collection of all random vectors identified by the corresponding lower case letter, up to and including the current time.

$$\boldsymbol{X}_k = \boldsymbol{x}_0, \boldsymbol{x}_1, \cdots, \boldsymbol{x}_k, \quad \boldsymbol{Y}_k = \boldsymbol{y}_1, \cdots, \boldsymbol{y}_k \tag{16}$$

Then, the BPF importance distribution is given by

$$\pi(\boldsymbol{x}_{k+1}|\boldsymbol{X}_k, \boldsymbol{Y}_{k+1}) = p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) \tag{17}$$

the sample $\boldsymbol{x}_{k+1}^{(i)}$ is obtained by first sampling $\boldsymbol{\nu}_k^{(i)}$ from the process noise. In this work the samples $\boldsymbol{\nu}_k^{(i)}$ are drawn from a Gaussian distribution and $\boldsymbol{x}_{k+1}^{(i)}$ are obtained as follows:

$$\boldsymbol{x}_{k+1}^{(i)} = \boldsymbol{f}_k(\boldsymbol{x}_k^{(i)}, \boldsymbol{\nu}_k^{(i)}) \tag{18}$$

The importance weights are calculated as

$$\omega_{k+1}^{(i)} \propto \omega_k^{(i)}\, p(\boldsymbol{y}_{k+1}|\boldsymbol{x}_{k+1}^{(i)}) = \omega_k^{(i)}\, n\Big(\boldsymbol{y}_{k+1};\ \boldsymbol{h}_k\big(\boldsymbol{x}_{k+1}^{(i)}\big),\ \boldsymbol{R}_{k+1}\Big) \tag{19}$$

Sample impoverishment is common in the BPF, and the weights update step is usually followed by a resampling step.

III. Sequential Monte Carlo Filtering with Gaussian Mixture Sampling

At each time step, Sequential Monte Carlo methods in general, and Particle Filters in particular, necessitate to start from a good set of samples that accurately and sufficiently represent the true distribution. Assuming such an initial set of samples exists, our goal is to approximate the distribution of $\boldsymbol{x}_k|\boldsymbol{Y}_k$ using sequential Monte Carlo methods. We will present one main algorithm and then show two small modifications to it. In a Monte Carlo method, ideally we would want to sample from $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$, and the first algorithm we propose does exactly that.

We assume that we have a good representation of the distribution at the prior time, that is to say, we have a set of of N i.i.d. samples $\boldsymbol{x}_{k-1}^{(i)}$ such that

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} \delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) \tag{20}$$

the basic idea behind the proposed algorithm is that the Dirac delta function $\delta(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k)$ is the limit as the covariance matrix goes to zero of a Gaussian distribution with mean $\bar{\boldsymbol{x}}_k^{(i)}$

$$\delta(\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) = n(\boldsymbol{x}_k;\ \bar{\boldsymbol{x}}_k, \boldsymbol{O}) \tag{21}$$

A. Algorithm I - Sampling from a GMM Posterior

As mentioned above we start from N i.i.d. samples of $p(\boldsymbol{x}_{k-1}|\boldsymbol{y}_{k-1})$ and we interpret the discretized distribution as a GMM

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N}\delta(\boldsymbol{x}_{k-1} - \boldsymbol{x}_{k-1}^{(i)}) = \sum_{i=1}^{N} \frac{1}{N} n(\boldsymbol{x}_{k-1}; \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{O}) \tag{22}$$

we can therefore propagate this distribution forward in time with the GSF equations to obtain

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N}\ n(\boldsymbol{x}_k; \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}), \boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}) \tag{23}$$

We can then process the measurement to obtain

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \approx \sum_{i=1}^{N} \omega_k^{(i)}\ n(\boldsymbol{x}_k; \boldsymbol{\mu}_k^{(i)}, \boldsymbol{P}_k^{(i)}) \tag{24}$$

$$\boldsymbol{\mu}_k^{(i)} = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}) + \boldsymbol{K}_k\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}))\Big) \tag{25}$$

$$\boldsymbol{P}_k^{(i)} = \boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}} - \boldsymbol{K}_k^{(i)}\boldsymbol{W}_k^{(i)}\boldsymbol{K}_k^{(i)\mathrm{T}} \tag{26}$$

$$\boldsymbol{K}_k^{(i)} = \boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}\ \boldsymbol{H}_k^{(i)\mathrm{T}}\ (\boldsymbol{W}_k^{(i)})^{-1} \tag{27}$$

$$\boldsymbol{W}_k^{(i)} = \boldsymbol{H}_k^{(i)}\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}\boldsymbol{H}_k^{(i)\mathrm{T}} + \boldsymbol{R}_k \tag{28}$$

$$\boldsymbol{G}_{k-1}^{(i)} = \left.\frac{\partial \boldsymbol{f}_{k-1}(\boldsymbol{x}, \boldsymbol{\nu})}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}=\boldsymbol{\nu}_{k-1}^{(i)}} \tag{29}$$

$$\boldsymbol{H}_k^{(i)} = \left.\frac{\partial \boldsymbol{h}_k(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{f}_{k-1}\left(\boldsymbol{x}_{k-1}^{(i)}\right)} \tag{30}$$

$$\omega_k^{(i)} \propto n\Big(\boldsymbol{y}_k;\ \boldsymbol{h}_k(\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)})),\ \boldsymbol{W}_k^{(i)}\Big) \tag{31}$$

where the weights in Eq. (31) are normalized.

We can now sample from the GMM distribution in Eq. (24) to obtain N i.i.d. samples of $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$; from these samples we can construct a Bayesian estimate and we can use them as a starting point for the next iteration.

To draw from a GMM we follow these steps:

1. Draw N samples $u^{(i)}$ from a uniform distribution between 0 and 1

2. For each i, find the index ℓ_i (where the subscipt i is to reinforce the fact there is one index for each value of $i = 1...N$) such that $\sum_{j=1}^{\ell_i - 1} \omega^{(j)} < u^{(i)} \leq \sum_{j=1}^{\ell_i} \omega^{(j)}$, where we define $\sum_{j=1}^{0} \omega^{(i)} = 0$

3. Draw $\boldsymbol{x}_k^{(i)}$ from the Gaussian distribution $n(\boldsymbol{x}_k;\ \boldsymbol{m}_{k|k-1}^{(\ell_i)}, \boldsymbol{P}_{k|k-1}^{(\ell_i)})$

Our approach of sampling from the GMM has two benefits. First, components with small weight are unlikely to produce a sample, therefore the resampling step is effectively already included in the sampling step. In the Bootstrap particle filter the process noise provides sample diversity after resampling. In a GMM the sample diversity is obtained directly since the Gaussian components are continuous distributions that already contain the contribution of the process noise. Other algorithms such as the regularized particle filter need to perform additional steps starting from the discrete distribution to obtain a continuous distribution to resample from. The second benefit of this algorithm is that, unlike the bootstrap particle filter, the GMM distribution accounts for the value of the measurement $\boldsymbol{y}_k$. This approach is reminiscent of the auxiliary particle filter, except that the full Bayes update is performed which allows us to directly sample rather than doing importance sampling.

Our proposed approach provides very good performance if:

1. The process noise covariance is not large enough such that the linearization of the measurement function $\boldsymbol{h}(\boldsymbol{x})$ is invalid in a region around $\boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{0}\big)$ whose spread is consistent with $\boldsymbol{G}_{k-1}^{(i)} \boldsymbol{Q}_{k-1} \boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$

2. The number of samples we start from is sufficient to accurately approximate the distribution at the prior time: $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N} n(\boldsymbol{x}_{k-1}; \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{O})$

This second assumption is common to all particle filters. If one of these two assumptions fail, the same algorithm proposed here can be used with the following mitigation strategies: 1) expressing the process noise itself as a GMM such that each component has a small enough covariance and 2) drawing more points from $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})$ as a starting point of the algorithm. However, at the cost of more computations, it is possible to mitigate these two issues in alternative ways as presented in the following two subsections.

B. Algorithm II - Importance Sampling from a GMM Posterior

If the distribution at the prior time is accurate, but we have reason to believe our GMM approximation of the distribution at the current time in Eq. (24) is not as accurate, it is possible to draw from the GMM in Eq. (24) as an importance distribution rather than as the true posterior. When drawing samples $\boldsymbol{x}_k^{(i)}$ from an importance distribution $\pi(\boldsymbol{x}_k|\boldsymbol{Y}_k)$ it is necessary to compute the true probability density $p(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k)$ in order to compute the importance weights. Therefore, we still need good knowledge of $p(\boldsymbol{x}_{k-1}^{(i)}|\boldsymbol{Y}_{k-1})$. Algorithm II proposed in this subsection provides a good methodology when we have enough samples $\boldsymbol{x}_{k-1}^{(i)}$ to accurately represent $p(\boldsymbol{x}_{k-1}^{(i)}|\boldsymbol{Y}_{k-1})$, but the GMM approximation of $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$ is not sufficiently accurate. This situation can occur when the linearization assumption taken by each of the components of the GMM is not accurate, such as when the nonlinearities of the measurement function $\boldsymbol{h}_k(\boldsymbol{x}_k)$ are significant in a region around $\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{0})$ spanned by the likely realizations of the component.

Assume the following approximation of the true posterior is more accurate than Eq. (24):

$$p(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k) \propto p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) \int p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1})\, p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})\, d\boldsymbol{x}_{k-1} \tag{32}$$

$$\approx p(\boldsymbol{y}_k|\boldsymbol{x}_k^{(i)}) \sum_{j=1}^{N} p(\boldsymbol{x}_k^{(i)}|\boldsymbol{x}_{k-1}^{(j)})\, p(\boldsymbol{x}_{k-1}^{(j)}|\boldsymbol{Y}_{k-1}) \tag{33}$$

$$= n\Big(\boldsymbol{y}_k;\ \boldsymbol{h}_k\big(\boldsymbol{x}_k^{(i)}\big), \boldsymbol{R}_k\Big) \sum_{j=1}^{N} \zeta_{k-1}^{(j)}\, n\Big(\boldsymbol{x}_k^{(i)};\ \boldsymbol{f}_{k-1}\big(\boldsymbol{x}_{k-1}^{(j)}\big), \boldsymbol{G}_{k-1}^{(j)} \boldsymbol{Q}_{k-1} \boldsymbol{G}_{k-1}^{(j)\mathrm{T}}\Big) \tag{34}$$

where the prior distribution $p(\boldsymbol{x}_{k-1}^{(j)}|\boldsymbol{Y}_{k-1})$ no longer has all weights equal to $1/N$:

$$p(\boldsymbol{x}_{k-1}^{(j)}|\boldsymbol{Y}_{k-1}) = \sum_{j=1}^{N} \zeta_{k-1}^{(j)}\, n\Big(\boldsymbol{x}_k^{(i)};\ \boldsymbol{x}_{k-1}^{(j)}, \boldsymbol{O}\Big) \tag{35}$$

then we can use Eq. (24) as the importance density

$$\pi(\boldsymbol{x}_k|\boldsymbol{Y}_k) = \sum_{i=1}^{N} \omega_k^{(i)}\, n\big(\boldsymbol{x}_k; \boldsymbol{\mu}_k^{(i)}, \boldsymbol{P}_k^{(i)}\big) \tag{36}$$

where $\omega_k^{(i)}$ are the weights of the i-th Gaussian component as defined in Eq. (31). The importance weights are calculated as

$$\xi_k^{(i)} = \frac{p(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k)}{\pi(\boldsymbol{x}_k^{(i)}|\boldsymbol{Y}_k)} \tag{37}$$

$$\zeta_k^{(i)} = \frac{\xi_k^{(i)}}{\sum_{i=1}^{N} \xi_k^{(i)}} \tag{38}$$

The posterior density is therefore approximated as

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \approx \sum_{i=1}^{N} \zeta_k^{(i)}\, \delta(\boldsymbol{x}_k - \boldsymbol{x}_k^{(i)}) = \sum_{i=1}^{N} \zeta_k^{(i)}\, n(\boldsymbol{x}_k; \boldsymbol{x}_k^{(i)}, \boldsymbol{O}) \tag{39}$$

notice that at the start of each iteration the initial weights $\zeta_k^{(i)}$ of the GMM are not $1/N$ as in Algorithm 1.

For large N, Algorithm II can be significantly more computationally expensive than Algorithm I because of the summation in Eq. (35) that is performed for each sample. In other words, Algorithm II has complexity of order N^2. Therefore in situations where the process noise covariance is large, expressing the process noise itself as a GMM and using Algorithm I is possibly preferable from a computational standpoint.

The pre-update error covariance matrix $\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$ being too small or not full rank could lead to particle impoverishment issues (all particle filters suffer from this problem). To overcome this, the following algorithm which uses nonzero initial covariances is proposed.

C. Algorithm III - Estimation with Non-Zero Initial Covariance

A better GMM approximation of $p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1})$ than Eq. (22) can be obtained by choosing nonzero covariances $\boldsymbol{P}_{k-1}^{(i)}$ for each of the components

$$p(\boldsymbol{x}_{k-1}|\boldsymbol{Y}_{k-1}) \approx \sum_{i=1}^{N} \frac{1}{N}\, n(\boldsymbol{x}_{k-1}; \boldsymbol{x}_{k-1}^{(i)}, \boldsymbol{P}_{k-1}^{(i)}) \tag{40}$$

Calculation of optimal values of $\boldsymbol{P}_{k-1}^{(i)}$ (for example minimizing the L2 norm of the difference between PDFs) is often infeasible or computationally expensive; a very simple alternative approach is to remove the bias in the sample covariance as described in this section.

When all the weights are the same, the covariance matrix of the GMM distribution is given by

$$\boldsymbol{P}_{k-1}^{GMM} = \frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{P}_{k-1}^{(i)} + \boldsymbol{x}_{k-1}^{(i)}\boldsymbol{x}_{k-1}^{(i)\mathrm{T}}\right) - \boldsymbol{\mu}_{k-1}\,\boldsymbol{\mu}_{k-1}^{\mathrm{T}} \tag{41}$$

$$\boldsymbol{\mu}_{k-1} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_{k-1}^{(i)} \tag{42}$$

when $\boldsymbol{P}_{k-1}^{(i)} = \boldsymbol{O}$ this reduces to

$$\begin{aligned}\boldsymbol{P}_{k-1}^{BIAS} &= \frac{1}{N}\sum_{i=1}^{N}\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{x}_{k-1}^{(i)}\,\boldsymbol{x}_{k-1}^{(i)\mathrm{T}}\right) - \boldsymbol{\mu}_{k-1}\,\boldsymbol{\mu}_{k-1}^{\mathrm{T}}\end{aligned} \tag{43}$$

which is a biased estimator of the covariance matrix since it is, on average, too small. An unbiased estimator of the covariance matrix is

$$\boldsymbol{P}_{k-1}^{UMB} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{44}$$

A very simple method to choose a nonzero value for the covariance matrix of the components in Eq. (40) is to choose $\boldsymbol{P}_{k-1}^{(i)}$ such that the GMM covariance matrix is unbiased

$$\begin{aligned}\frac{1}{N}\left(\sum_{i=1}^{N} \boldsymbol{P}_{k-1}^{(i)} + \boldsymbol{x}_{k-1}^{(i)}\boldsymbol{x}_{k-1}^{(i)\mathrm{T}}\right) - \boldsymbol{\mu}_{k-1}\,\boldsymbol{\mu}_{k-1}^{\mathrm{T}} \\ = \frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(i)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}}\end{aligned} \tag{45}$$

assuming all samples have the same covariance matrix, the solution to this equation is given by

$$\begin{aligned}\boldsymbol{P}_{k-1}^{(i)} &= \frac{1}{N(N-1)}\sum_{j=1}^{N}\left(\boldsymbol{x}_{k-1}^{(j)} - \boldsymbol{\mu}_{k-1}\right)\,\left(\boldsymbol{x}_{k-1}^{(j)} - \boldsymbol{\mu}_{k-1}^{\mathrm{T}}\right)^{\mathrm{T}} \\ &= \frac{1}{N}\boldsymbol{P}_{k-1}^{UMB} \qquad \forall i\end{aligned} \tag{46}$$

For example, if we have a scalar state with umbiased sample variance of $P_{k-1}^{UMB} = 1$ and we choose to draw 100 samples, each of the 100 components of the GMM will have standard deviation $\sigma_{k-1}^{(i)} = \sqrt{P_{k-1}^{(i)}} = 0.1$.

The remainder of the algorithm is similar to Algorithm I

$$p(\boldsymbol{x}_k|\boldsymbol{Y}_k) \approx \sum_{i=1}^{N}\omega_k^{(i)}\; n(\boldsymbol{x}_k; \boldsymbol{\mu}_k^{(i)}, \boldsymbol{P}_k^{(i)}) \tag{47}$$

$$\boldsymbol{\mu}_k^{(i)} = \boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}) + \boldsymbol{K}_k\Big(\boldsymbol{y}_k - \boldsymbol{h}_k(\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)}))\Big) \tag{48}$$

$$\boldsymbol{P}_k^{(i)} = \boldsymbol{F}_{k-1}^{(i)}\boldsymbol{P}_{k-1}^{(i)}\boldsymbol{F}_{k-1}^{(i)} + \boldsymbol{G}_{k-1}^{(j)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(j)\mathrm{T}} - \boldsymbol{K}_k^{(i)}\boldsymbol{W}_k^{(i)}\boldsymbol{K}_k^{(i)\mathrm{T}} \tag{49}$$

$$\boldsymbol{K}_k^{(i)} = (\boldsymbol{F}_{k-1}^{(i)}\boldsymbol{P}_{k-1}^{(i)}\boldsymbol{F}_{k-1}^{(i)} + \boldsymbol{G}_{k-1}^{(j)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(j)\mathrm{T}})\;\boldsymbol{H}_k^{(i)\mathrm{T}}\;(\boldsymbol{W}_k^{(i)})^{-1} \tag{50}$$

$$\boldsymbol{W}_k^{(i)} = \boldsymbol{H}_k^{(i)}(\boldsymbol{F}_{k-1}^{(i)}\boldsymbol{P}_{k-1}^{(i)}\boldsymbol{F}_{k-1}^{(i)} + \boldsymbol{G}_{k-1}^{(j)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(j)\mathrm{T}})\boldsymbol{H}_k^{(i)\mathrm{T}} + \boldsymbol{R}_k \tag{51}$$

$$\boldsymbol{F}_{k-1}^{(i)} = \left.\frac{\partial \boldsymbol{f}_{k-1}(\boldsymbol{x},\boldsymbol{\nu})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{k-1}^{(i)}} \tag{52}$$

$$\boldsymbol{G}_{k-1}^{(i)} = \left.\frac{\partial \boldsymbol{f}_{k-1}(\boldsymbol{x},\boldsymbol{\nu})}{\partial \boldsymbol{\nu}}\right|_{\boldsymbol{\nu}=\boldsymbol{\nu}_{k-1}^{(i)}} \tag{53}$$

$$\boldsymbol{H}_k^{(i)} = \left.\frac{\partial \boldsymbol{h}_k(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{f}_{k-1}\left(\boldsymbol{x}_{k-1}^{(i)}\right)} \tag{54}$$

$$\omega_k^{(i)} \propto n\Big(\boldsymbol{y}_k;\; \boldsymbol{h}_k(\boldsymbol{f}_{k-1}(\boldsymbol{x}_{k-1}^{(i)})),\; \boldsymbol{W}_k^{(i)}\Big) \tag{55}$$

the weights in Eq. (55) are normalized and we can now sample from this GMM distribution to obtain N i.i.d. samples of $p(\boldsymbol{x}_k|\boldsymbol{Y}_k)$.

In Algorithm III, we calculate the actual posterior distribution as a GMM and sample directly from it. Moreover, the covariance matrix of the components $\boldsymbol{P}_{k-1}^{(i)}$ is calculated, which makes Algorithm III practically and conceptually different from regularized particle filter in that the covariance is not merely used for particle resampling.

IV. Numerical Results

In order to evaluate the algorithms proposed in this paper, four different examples are considered: a simple motivating example, the univariate nonstationary growth model (used in [20, 25, 28]), a Lorenz96 system (used in [27, 29]), and the blind tricyclist problem (used in [23, 24, 30]).

A. Single Step Example

Consider the following simple motivating example. A bivariate normal random vector $\boldsymbol{x}_0$ is distributed as

$$\boldsymbol{x}_0 \sim n\left(\boldsymbol{x}_0;\ \boldsymbol{\mu}_0,\ \boldsymbol{P}_0\right) = n\left(\boldsymbol{x}_0;\ \begin{bmatrix} -3 \\ 0 \end{bmatrix},\ \begin{bmatrix} 7.2 & 0 \\ 0 & 21.6 \end{bmatrix}\right) \tag{56}$$

and evolves as

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{\nu} \tag{57}$$

where

$$\boldsymbol{\nu} \sim n\left(\boldsymbol{\nu};\ \boldsymbol{0},\ \boldsymbol{Q}\right) = n\left(\boldsymbol{\nu};\ \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}\right) \tag{58}$$

a measurement is available and given by

$$y = \|\boldsymbol{x}_1\| + \eta \tag{59}$$

where

$$\eta \sim n\left(\eta;\ 0,\ R\right) = n\left(\eta;\ 0,\ 0.01\right) \tag{60}$$

we start from $N = 300$ independently drawn samples of $\boldsymbol{x}_0$ and we apply the Bootstrap Particle Filter (BPF), the Auxiliary Particle Filter (APF), Algorithm 1 (A1) and Algorithm 3 (A3) from this paper.

For the BPF we draw N independent samples from $\boldsymbol{\eta}$, and update the weights as

$$w_{BPF}^{(i)} \propto n(\boldsymbol{\nu}^{(i)};\ \boldsymbol{0}, \boldsymbol{Q})\ n(\|\boldsymbol{x}_0^{(i)} + \boldsymbol{\nu}^{(i)}\|;\ 0, R) \tag{61}$$

the effective number of particles is calculated as

$$N_{eff}^{BPF} = \frac{1}{\sum_{i=1}^{N}(w_{BPF}^{(i)})^2} \tag{62}$$

and $\boldsymbol{x}_1^{(i),BPF} = \boldsymbol{x}_0^{(i)} + \boldsymbol{\nu}^{(i)}$ with associated weight $w_{BPF}^{(i)}$. After resampling, many resampled Bootstrap particles $\tilde{\boldsymbol{x}}_1^{(i),BPF}$ will coincide and all particles will have equal weight $1/N$.

For the APF resampling of the initial state is performed

$$\boldsymbol{x}_0 \approx \sum_{i=1}^{n} w_{APF}^{(i)}\, \delta(\boldsymbol{x}_0 - \boldsymbol{x}_0^{(i)}) \tag{63}$$

where

$$w_{APF}^{(i)} \propto n(\|\boldsymbol{x}_0^{(i)} + \boldsymbol{\nu}^{(i)}\|;\, 0, R) \tag{64}$$

the effective number of particles is calculated as

$$N_{eff}^{APF} = \frac{1}{\sum_{i=1}^{N}(w_{APF}^{(i)})^2} \tag{65}$$

and $\boldsymbol{x}_1^{(i),APF} = \tilde{\boldsymbol{x}}_0^{(i)} + \boldsymbol{\nu}^{(i)}$ with associated weight $1/N$ where $\tilde{\boldsymbol{x}}_0^{(i)}$ are resampled particles. Notice that because of $\boldsymbol{\nu}^{(i)}$, all particles $\boldsymbol{x}_1^{(i),APF}$ are distinct from one another.

For A1 and A3 (jointly denoted as AN) we use the weights described in Eq. (31) and Eq. (55), and we calculate the effective number of particles as

$$N_{eff}^{AN} = \frac{1}{\sum_{i=1}^{N}\left(\omega_{AN}^{(i)}\right)^2} \tag{66}$$

After sampling from the GMM, all sampled particles $\boldsymbol{x}_1^{(i),AN}$ are distinct from one another and have weight $1/N$.

Performing 100 random experiments for each of the four filters, we obtain the average number of effective particles,the root mean square error (RMSE) and the Cramer-Rao lower bound (CRLB) [30, 31] values given in Table 1. The results show that starting from the same initial 300 particles the proposed methodologies produce the most sample diversity and best accuracy among the filters. Notice that none of the algorithms in Table 1 (new or existing) approach the Cramer-Rao lower bound, this is true for the following examples as well. This is due to the complex nonlinear nature of the examples chosen.

Ex. 1	Effective Particles	RMSE
BPF (300)	9.8370	1.8215
APF (300)	12.4878	1.7847
A1 (300)	56.3863	1.6333
A3 (300)	62.0461	1.6228
CRLB		0.1999

Table 1 Results of Example 1: Number of Effective Particles

B. Univariate Nonstationary Growth Model

Consider the discrete time highly nonlinear scalar dynamic system and measurement model given by [20, 25, 28]:

$$x_k = \frac{1}{2}x_{k-1} + 25\frac{x_{k-1}}{1+x_{k-1}^2} + 8\cos\big(1.2(k-1)\big) + \nu_{k-1} \tag{67}$$

$$y_k = \frac{x_k^2}{20} + \eta_k \tag{68}$$

where the process noise, ν_{k-1}, and the measurement noise, η_k, are assumed to be independent zero mean Gaussian random variables with variances $Q = 1$ and $R = 1$, respectively.

This model is highly nonlinear and bimodal. The cosine term in the dynamic equation varies with time k. The likelihood has a bimodal nature which makes the states more difficult to estimate. In this example, a Monte Carlo analysis is performed with 200 simulations, each simulation has a time span $k = [0,\ 50]$. The estimation performance of the EKF, UKF, Bootstrap PF (BPF) and the three algorithms proposed here (A1, A2, A3) are compared based on RMSE, effective sample size (ESS), and noncredibility index (NCI) [32]. The RMSE for each Monte Carlo simulation is calculated from the true and estimated states at each time k. The ESS is the effective number of particles calculated as in the previous example. The NCI is defined as

$$\mathrm{NCI}_k = \frac{1}{M}\sum_{j=1}^{M}\Big[10\log_{10}\Big((\boldsymbol{x}_k^j-\boldsymbol{\mu}_k^j)^{\mathrm{T}}(\boldsymbol{P}_k^j)^{-1}(\boldsymbol{x}_k^j-\boldsymbol{\mu}_k^j)\Big) - 10\log_{10}\Big((\boldsymbol{x}_k^j-\boldsymbol{\mu}_k^j)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}_k^j-\boldsymbol{\mu}_k^j)\Big)\Big] \tag{69}$$

where M is the number of Monte Carlo simulations, $\boldsymbol{x}_k^j$ are the true states, $\boldsymbol{\mu}_k^j$ are the estimated states, $\boldsymbol{P}_k^j$ are the filter's error covariance matrix of the j-th Monte Carlo run computed with Eq. (14), and $\boldsymbol{\Sigma}_k$ is the ensemble error covariance matrix of the estimates at time k computed from the Monte Carlo samples.

The NCI quantifies the difference between the ideal error covariance matrix $\boldsymbol{\Sigma}_k$ and the estimated error covariance matrix $\boldsymbol{P}_k$. The NCI metric is a geometric average of 10 times the logarithm of the normalized estimation error squared (NEES) ratio; it is a balanced measure of the consistency of the estimators. When the difference between $\boldsymbol{\Sigma}_k$ and $\boldsymbol{P}_k$ is small, the NCI value should be zero or nearly zero at all times [32].

The Root Mean Square of the RMSEs, the Monte Carlo averaged ESS, and the NCI from the 200 Monte Carlo runs are shown in Table 2. A total of 100 particles are used in both the BPF and the new algorithms proposed here.

Fig. 1 shows the RMSE and the CRLB of the 200 simulation, the RMSE for each is calculated over a time span of $[0,\ 50]$. The RMSE values of each filter are listed in Table 2. Our three proposed algorithms have comparable RMSEs. The best performance is obtained with A3, which starts each iteration from a GMM with non-zero covariance. The RMSEs of the EKF and UKF are higher than that of any sample-based filters. Moreover, the proposed algorithms have better performance than the BPF given the same number of particles, 100.

The consistency test result of each estimator represented by the absolute NCI value is depicted in Fig. 2. In this figure, the NCI values of our proposed algorithms are smaller than those of other estimators. Fig. 3 describes the ESS which indicates sample diversity of particle filters. In the figure, the Monte Carlo simulations show that the proposed methodologies produce significantly higher effective number of particles than the BPF. The proposed A3 method performs best in both ESS while A2 is slightly more consistent.

Ex. 2	**RMSE**	**ESS**	**NCI**
A1 (100)	22.6558	78.0530	5.3895
A2 (100)	22.4748	77.6704	4.1806
A3 (100)	22.3328	79.0343	4.1823
BPF (100)	23.5081	60.2690	6.0290
EKF	71.7314		17.2206
UKF	50.2221		10.3616
CRLB	1.7258		

Table 2 Results of Example 2: RMSE for 200 Monte Carlo Simulations

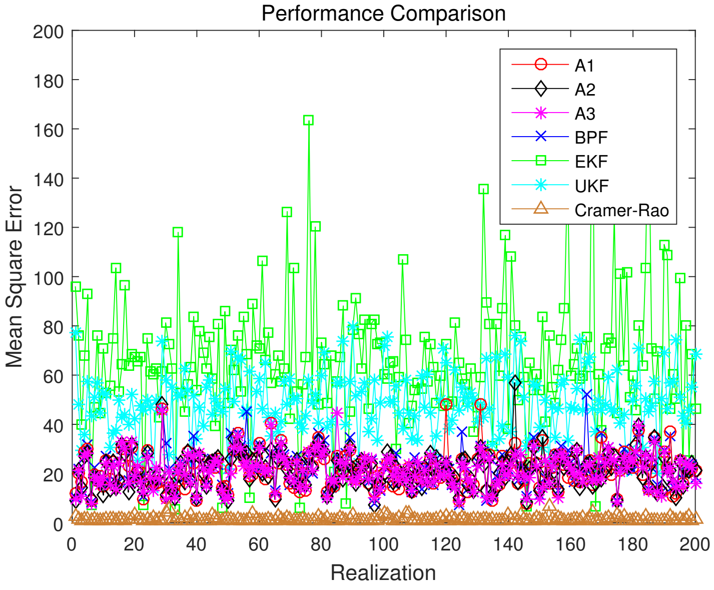

Fig. 1 Time averaged RMSE for 200 Random Realizations

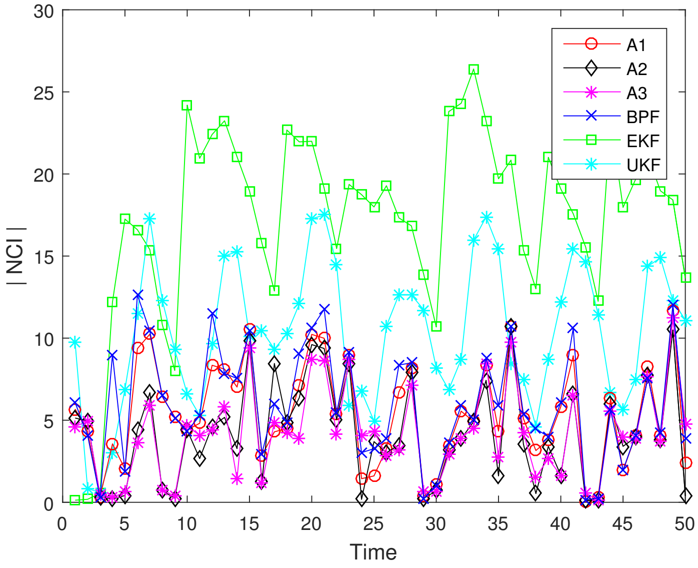

Fig. 2 Time history of the absolute NCI value for 200 Random Realizations

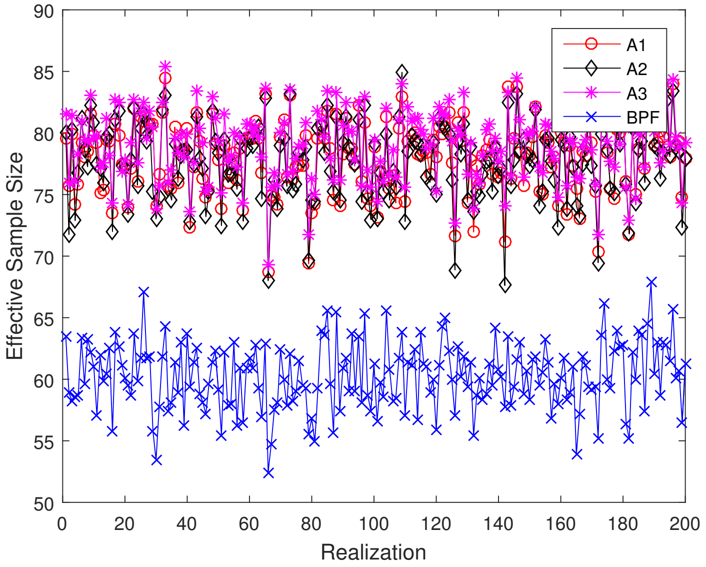

Fig. 3 Time averaged effective sample size for 200 Random Realizations

C. Lorenz96 system

In this example, the Bootstrap PF and the here proposed Algorithm 1 and Algorithm 3 are applied to a Lorenz96 system [27, 29]. The Lorenz96 dynamical system is expressed as follows:

$$\dot{x}_i(t) = x_{i-1}(t)\left(x_{i+1}(t) - x_{i-2}(t)\right) - x_i(t) + F + \nu_i(t) \tag{70}$$

$$y_k = H\ X(t_k) + \eta_k, \quad H_{i,j} = \begin{cases} 1,\ j = 2i-1 \\ 0,\ \text{otherwise} \end{cases},\ \text{for } i = 1, \cdots, 20,\ j = 1, \cdots, 40 \tag{71}$$

where $x_i(t)$, $i = 1, 2, \cdots, 40$, are the components of the 40th-dimensional vector $X(t)$. In the dynamics equation the following conventions are used $x_{-1} = x_{N-1}$, $x_0 = x_N$, and $x_1 = x_{N+1}$. The term F represents a constant external forcing and is set to 8, which causes chaotic behavior in the system. The dynamics is propagated for 10 seconds at 20 Hz while the discrete measurements are available at 1 Hz, $t_k = 1, 2, \cdots, 200$. Fourth order Runge-Kutta integration is used with a step size of 0.05 sec, and the process noise is held constant over each 0.05 second interval with zero correlation between the intervals. The measurements are linear and measure only the components of the state vector that have odd indices. It is assumed that the process noise and measurement noise are uncorrelated, white, zero mean, and with covariance matrices given by $Q = 10^{-2}$ and $R = 10^{-2}\boldsymbol{I}_{20\times 20}$, respectively [27]. The initial state of the system is assumed multivariate Gaussian distribution with $\boldsymbol{\mu}_0 = F[1, 1, \cdots, 1]^{\mathrm{T}}$ and $P_0 = 10^{-3}\boldsymbol{I}_{40\times 40}$.

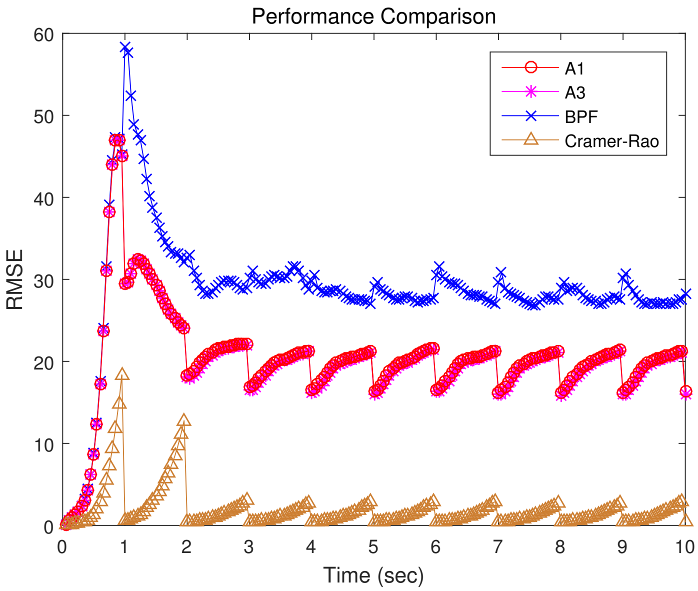

Fig. 4 Monte Carlo averaged RMSE for 100 Random Realizations

Fig. 4 shows the CRLB and the performance of 100 Monte Carlo simulations with 2000 particles for the Lorenz96 system. The time averaged value of RMSE of the three algorithms and the CRLB are shown in Table 3. For such large system, A2 is not recommended because of high computation time and is omitted from this example. The results show that the performance of A3 is better than the A1. Moreover, the BPF is found to provide significantly inferior performance. In order to compare the consistency of the filters, the absolute NCI value is computed and compared in Fig. 5. This figure indicates that the performance of A1 and A3 are comparable. On the other hand, the absolute NCI value of the BPF is greater than that of A1 and A3 over time. The time averaged ESS for the 100 Monte Carlo simulations are shown in Fig. 6. The effective number of particles for the BPF is small since it does not directly account for the latest information of the measurement. On the other hand, A1 and A3 provide good diversity and number of effective particles. The quantitative results representing the consistency and ESS of the filters are listed in Table 3. A3 has the best performance in terms of accuracy, consistency, and ESS.

D. The Blind Tricyclist Problem

In this last example, Algorithm 3 is tested on the blind tricyclist problem presented in Ref. [30], and its performance is compared to that of an EKF and Regularized Particle Filter (RPF). The blind tricyclist is a challenging nonlinear estimation problem with seven states consisting of unknown planar position, heading

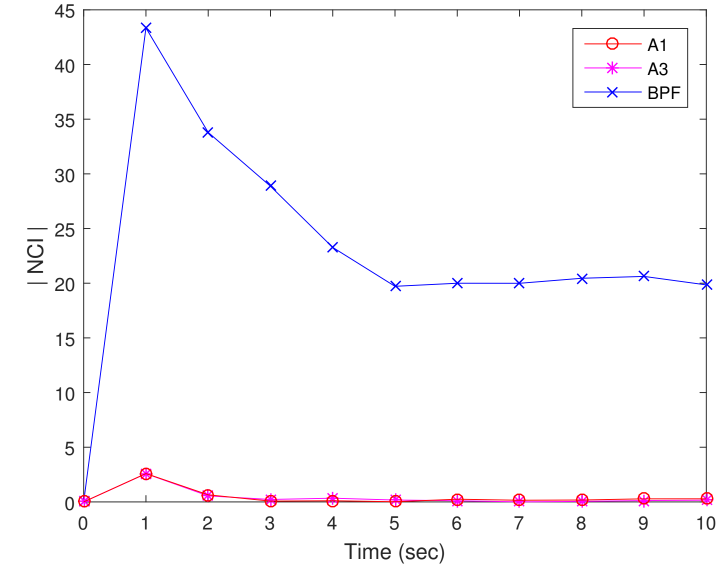

Fig. 5 Time history of the absolute NCI value for 100 Random Realizations

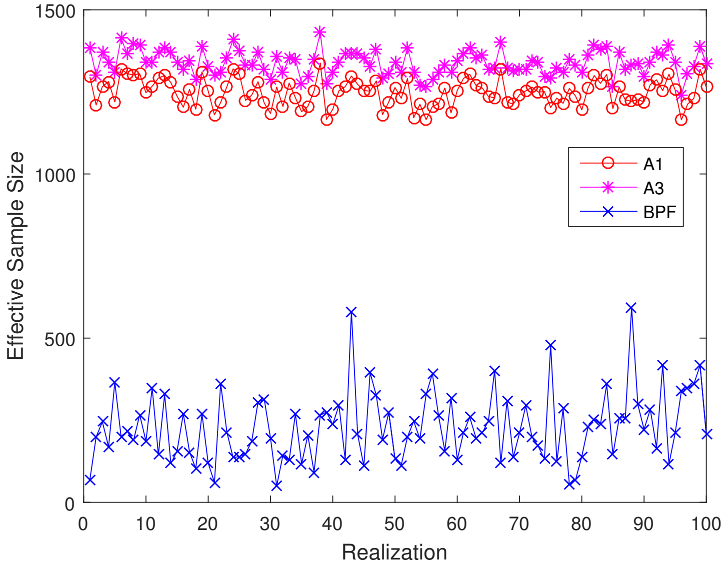

Fig. 6 Time averaged effective sample size for 100 Random Realizations

angle, and four observation parameters. Unlike the previous examples, in this problem the process noise does not enter the dynamics linearly. Moreover, the process noise covariance matrix is not full rank because three states do not have process noise. Therefore, most particle filters will fail to produce particle diversity, while A3 and RPF are suitable and applied to this problem. The dynamics are propagated for 141 seconds at 2 Hz with the two known inputs corrupted by additive Gaussian noise. Two relative bearing measurements are

Ex. 3	RMSE	ESS	NCI
A1 (2000)	20.3819	1248.5274	0.4160
A3 (2000)	20.1415	1338.7312	0.3920
BPF (2000)	28.7860	229.0955	22.7186
CRLB	1.8377		

Table 3 Results of Example 3: RMSE for 100 Monte Carlo Simulations

available every 3 seconds out-of-phase at 180 ° E.g., the rider gets relative bearing measurements from two shouting friends: the first friend shouts out at sample times 0.5, 3.5, 6,5, etc., while the second friend shouts out at sample times 2, 5, 8, etc.

Fig. 7 displays the time history of the position's RMSE magnitude of the CRLB and 100 Monte Carlo simulations of an EKF, A3 with 3000 and 10000 particles, and RPF with 3000 and 10000 particles. Since the process noise is only related to the planar position and heading states, the process noise covariance matrix $\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$ is not full rank. Therefore, A1 and A2 cannot be successfully applied to this problem, neither are BPF and APF. In addition, since the tricycle heading angle and the merry-go-round phase angles can cause a 2π cycle ambiguity, a 2π relative unwrapping operation is performed. The RPF resampling is done whenever the number of effective particles is smaller than a resampling threshold $\hat{N}_{eff}$, chosen as 400 and 5000 for 3000 and 10000 particles, respectively [24]. The results indicate that the performances of both the A3s with 3000 and 10000 mixture elements are better than those of the RPFs for the first 100 sec but they become comparable after that. The reason is that RPF implemented here uses an Epanechnikov kernel density estimator, which is optimal for Gaussian distributions. After 100 seconds of simulation time, when the total uncertainty of the problem reduces, the distribution looks "more" Gaussian and the RPF performs really well. However, when the PDF differs substantially from Gaussian, the Epanechnikov kernel density estimator and hence the RPF perform noticeably worse than A3. If the posterior density was known, an optimal kernel estimator could be found to produce excellent results. Generally speaking, however, the shape of the posterior distribution is unknown and thus A3 does a better job of representing the distribution, as the consistency test below clearly shows. Ref. [30] details the reason why RPF with 10000 particles performs worse than the RPF with 3000 particles: "First, the increase from 3000 to 10,000 particles might be insufficient to ensure improvement in a 100-run Monte Carlo simulation. Also, the RPF regularization?s dithering might have interfered with the PF?s accuracy convergence in the limit of a large number of particles."

The RMSE of A3 and RPF lie a bit lower than the CRLB during the first 5 sec of the run, which is theoretically impossible but allowable since a finite number of Monte Carlo simulations is conducted [23]. This figure also shows that the performance of the EKF is inferior to that of A3 and RPF. The quantitative RMSE results for position is listed in Table 4. The RMSE value of the A3 with 3000 particles is 3.97% smaller than that of the RPF with 3000 particles and the RMSE value of the A3 with 10000 particles is 27.87% smaller than that of the RPF with 10000 particles.

Fig. 8 shows the absoulte NCI value of each estimator. The absolute value of NCI of all filters increases as time passes. This is because the process noise covariance matrix $\boldsymbol{G}_{k-1}^{(i)}\boldsymbol{Q}_{k-1}\boldsymbol{G}_{k-1}^{(i)\mathrm{T}}$ is rank-deficient. It is well known that small process noise can causes degeneracy in particle filters, thus degrading their performance [19]. The figure shows that the RPF with 3000 particles does suffer from degeneracy. Even with 3000 particles, the absolute value of NCI of A3 shows that the filter is performing in a very satisfactory fashon. The time averaged absolute NCI value to the total samples of 100 cases is listed in Table 4, where n/a indicates degeneracy. The average computation time per filtering run in MATLAB on a 3.5-GHz, four-core Ubuntu operation system is also presented in Table 4. The absolute NCI value of the A3 is smaller than that of the EKF and RPF with the same number of particles. In addition, compared to the RPF, the A3 reduces the mean computation time by 5.65% and 19.72% with 3000 and 10000 particles, respectively. Therefore, the performance in terms of accuracy, consistency, and mean computation time of the proposed algorithm is conspicuously better than that of the EKF and RPF.

V. Conclusions

In this paper, a new sequential Monte Carlo algorithm is proposed that samples from a Gaussian Mixture Model (GMM) approximation of the posterior distribution. Each sample of the distribution at the prior time is treated as a Gaussian component with a collapsed zero covariance matrix. Process noise is responsible for generating propagated components with non-singular covariance matrix, and the Gaussian Sum Filter algorithm is used to calculate the posterior distribution. It is shown that the proposed algorithm improves over the accuracy, consistency, and effective number of particles of the Bootstrap and Regularized Particle

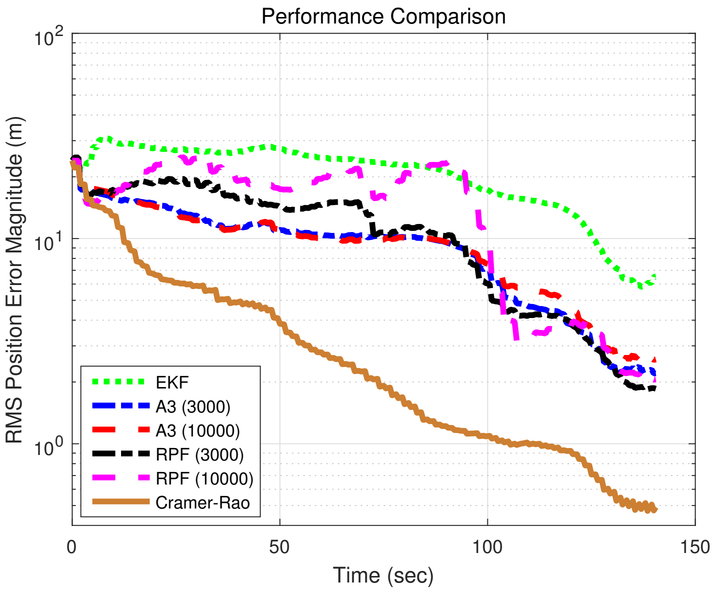

Fig. 7 Monte Carlo averaged RMSE for 100 Random Realizations

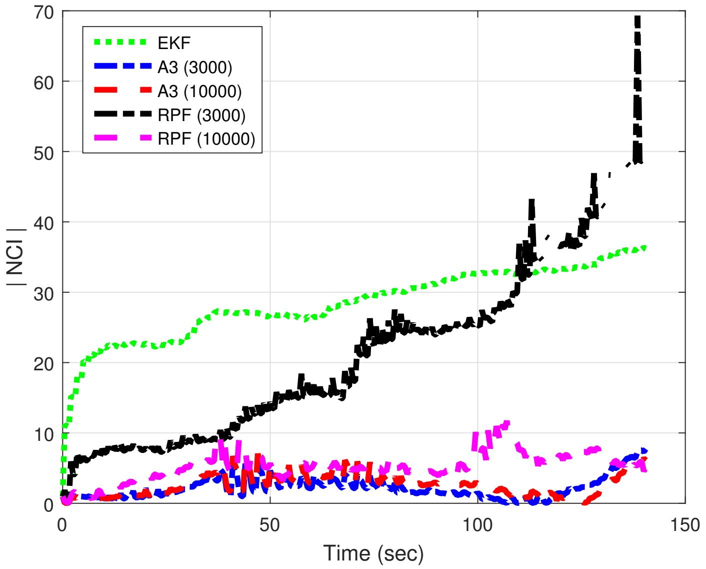

Fig. 8 Time history of the absolute NCI value for 100 Random Realizations

Filters in the numerical examples considered.

Two small modifications of the baseline algorithm are also proposed to further improve its accuracy. First, an importance sampling version of the algorithm is developed. At the cost of more computations, this modified approach slightly improves over the baseline algorithm. In the second modification, the initial covariance of the GMM components is not set to zero, but to a small value that removes the bias in the sample

Ex. 4	RMSE	NCI	Computation time (sec/sim.)
EKF	21.2319	28.3933	0.0205
A3 (3000)	11.0344	2.1958	41.3793
A3 (10000)	10.8863	2.6238	170.2232
RPF (3000)	11.4904	n/a	43.8562
RPF (10000)	15.0916	5.3220	212.0285
CRLB	3.9161		

Table 4 Results of Example 4: RMSE for 100 Monte Carlo Simulations

covariance, this approach is necessary, for example, when the process noise is not sufficient to produce non-singular covariance matrices for the components. All the proposed algorithms have better performance than the conventional Bootstrap Particle Filter in all tests performed.

VI. Aknowledgement

This work was sponsored in part by the Air Force Office of Scientific Research under grant number FA9550-18-1-0351. We would like to thank Dr. Mark Psiaki for his help in sharing his simulation and code for the blind tricyclist example.

VII. References

[1] Doucet, A., de Freitas, N., and Gordon, N. J., *Sequential Monte Carlo Methods in Practice*, Springer, New York, 2001, ISBN: 978-1-4757-3437-9.

[2] Arulampalam, S., Gordon, N., and Ristic, B., *Beyond the Kalman Filter: Particle Filters for Tracking Applications Partical Filters for Tracking Applications*, Artech House, 2004, ISBN: 978-1-5805-3631-8.

[3] Sarkka, S., *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, doi: 10.1017/CBO9781139344203.

[4] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, No. Series D, March 1960, pp. 35–45, doi: 10.1115/1.3662552.

[5] Gelb, A., editor, *Applied Optimal Estimation*, The MIT press, Cambridge, MA, 1974, ISBN: 978-0-2625-7048-0.

[6] Lefebvre, T., Bruyninckx, H., and Schutter, J. D., "Comment on "A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators"," *IEEE Transactions on Automatic Control*, Vol. 47, No. 8,

2002, pp. 1406–1408, doi: 10.1109/TAC.2002.800742.

[7] Arasaratnam, I., Haykin, S., and Elliot, R. J., “Discrete-Time Nonlinear Filtering Algorithms using Gauss-Hermite Quadrature,” *Proceedings of the IEEE*, Vol. 95, No. 5, May 2007, pp. 953–977, doi: 10.1109/JPROC.2007.894705.

[8] Julier, S. J. and Uhlmann, J. K., “New Extension of the Kalman Filter to Nonlinear Systems,” *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls, Orlando, FL*, 1997, doi: 10.1117/12.280797.

[9] Arasaratnam, I. and Haykin, S., “Cubature Kalman Filters,” *IEEE Transactions on Automatic Control*, Vol. 54, No. 6, June 2009, pp. 1254 – 1269, doi: 10.1109/TAC.2009.2019800.

[10] Sorenson, H. W. and Alspach, D. L., “Recursive Bayesian Estimation Using Gaussian Sums,” *Automatica*, Vol. 7, No. 4, July 1971, pp. 465–479, doi:10.1016/0005-1098(71)90097-5.

[11] Alspach, D. and Sorenson, H., “Nonlinear Bayesian estimation using Gaussian sum approximations,” *IEEE Transactions on Automatic Control*, Vol. 17, No. 4, August 1972, pp. 439–448, doi: 10.1109/SAP.1970.270017.

[12] DeMars, K. J., Bishop, R. H., and Jah, M. K., “An entropy-based approach for uncertainty propagation of nonlinear dynamical systems,” *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, July–August’ 2013, pp. 1047–1057, doi: 10.2514/1.58987.

[13] Faubel, F., McDonough, J., and Klakow, D., “The split and merge unscented Gaussian mixture filter,” *IEEE Signal Processing Letters*, Vol. 16, No. 9, 2009, pp. 786–789, doi: 10.1109/LSP.2009.2024859.

[14] Faubel, F. and Klakow, D., “Further improvement of the adaptive level of detail transform: Splitting in direction of the nonlinearity,” *Signal Processing Conference, 2010 18th European*, IEEE, 2010, pp. 850–854, ISSN: 2219-5491.

[15] Vittaldev, V. and Russell, R. P., “Collision probability for space objects using Gaussian mixture models,” *Proceedings of the 23rd AAS/AIAA Space Flight Mechanics Meeting*, Vol. 148, Advances in the Astronautical Sciences, Univelt San Diego, CA, 2013, pp. 2339–2358, ISBN: 978-0-8770-3597-8.

[16] Havlak, F. and Campbell, M., “Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments,” *IEEE Transactions on Robotics*, Vol. 30, No. 2, 2014, pp. 461–474, doi: 10.1109/TRO.2013.2291620.

[17] Raitoharju, M. and Ali-Loytty, S., “An adaptive derivative free method for Bayesian posterior approximation,” *IEEE Signal Processing Letters*, Vol. 19, No. 2, 2012, pp. 87–90, doi: 10.1109/LSP.2011.2179800.

[18] Tuggle, K. and Zanetti, R., “Automated Splitting Gaussian Mixture Nonlinear Measurement Update,” *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 3, March 2018, pp. 725–734, doi: 10.2514/1.G003109.

[19] Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T., “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on Signal Processing*, Vol. 50, No. 2, 2002, pp. 174–188, doi: 10.1109/78.978374.

[20] Gordon, N. J., Salmond, D. J., and Smith, A. F. M., “Novel approach to nonlinear/non-Gaussian Bayesian state

estimation," *Radar and Signal Processing, IEE Proceedings F*, Vol. 140, No. 2, April 1993, pp. 107–113, doi: 10.1049/ip-f-2.1993.0015.

[21] Pitt, M. and Shephard, N., "Filtering via simulation: auxiliary particle," *J. Am. Statist. Ass.*, , No. 94, 1999, pp. 590–599, doi: 10.1080/01621459.1999.10474153.

[22] Schrempf, O. C., Brunn, D., and Hanebeck, U. D., "Density approximation based on dirac mixtures with regard to nonlinear estimation and filtering," *Proceedings of the 2006 IEEE Conference on Decision and Control (CDC 2006)*, San Diego, California, 2006, doi: 10.1109/CDC.2006.376759.

[23] Psiaki, M. L., Schoenberg, J. R., and Miller, I. T., "Gaussian Sum Reapproximation for Use in a Nonlinear Filter," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 2, 2015, pp. 292–303, doi: 10.2514/1.G000541.

[24] Psiaki, M. L., "Gaussian Mixture Nonlinear Filtering With Resampling for Mixand Narrowing," *IEEE Transactions on Signal Processing*, Vol. 64, No. 21, 2016, pp. 5499–5512, doi: 10.1109/TSP.2016.2595503.

[25] Kotecha, J. H. and Djuric, P. M., "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, Vol. 51, No. 10, Oct 2003, pp. 2602–2612, doi: 10.1109/TSP.2003.816754.

[26] Kotecha, J. H. and Djuric, P. M., "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, Vol. 51, No. 10, Oct 2003, pp. 2592–2601, doi: 10.1109/TSP.2003.816758.

[27] Raihan, D. and Chakravorty, S., "Particle Gaussian Mixture (PGM) filters," July 2016, pp. 1369–1376, ISBN: 978-0-9964-5274-8.

[28] Kitagawa, G., "Non-Gaussian state-space modeling of nonstationary time series (with discussion)," *Journal of the American Statistical Association*, Vol. 82, No. 400, 1987, pp. 1032–1063, doi: 10.2307/2289375.

[29] Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D., and Yorke, J. A., "A Local Ensemble Kalman Filter for Atmospheric Data Assimilation," *Tellus A: Dynamic Meteorology and Oceanography*, Vol. 56, No. 5, 2004, pp. 415–428, doi: 10.3402/tellusa.v56i5.14462.

[30] Psiaki, M. L., "The blind tricyclist problem and a comparative study of nonlinear filters," *IEEE Control Systems Magazine*, Vol. 33, No. 3, 2013, pp. 40–54, doi: 10.1109/MCS.2013.2249422.

[31] Tichavsky, P., Muravchik, C. H., and Nehorai, A., "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, 1998, pp. 1386–1396, doi: 10.1109/78.668800.

[32] Li, X. R. and Zhao, Z., "Measuring Estimator's Credibility: Noncredibility Index," July 2006, doi: 10.1109/I-CIF.2006.301770.